Network Intrusion Detection Using Snort - ICMP and HTTP Traffic Detection

Platform: Ubuntu on VirtualBox
Date: [10th April 2025]
Name: Shraddha Dinesh Chavan/CA/MA1/10080

What is NIDS?

A security device known as a Network Intrusion Detection System (NIDS) is developed to monitor and evaluate network traffic for unusual activity, illegal access, or policy violations. It collects data packets as they go over a network and rapidly detects potential vulnerabilities.

Why Use Snort in NIDS?

Snort is one of the most widely used open-source tools for building a NIDS. It offers:

Packet sniffing and logging capabilities

Custom rule creation to define specific traffic patterns to detect

Real-time alerts when suspicious traffic is detected

Flexibility to work in small lab setups or large enterprise networks

1. Introduction

This report documents the process of setting up and testing a Network Intrusion Detection System (NIDS) using Snort to detect ICMP and HTTP traffic. The objective was to create custom rules and verify real-time detection between two virtual machines in a controlled lab environment.

2. Tools Used

-Snort (v3) on Ubuntu
- Apache2 Web Server (for HTTP testing)
- Kali Linux (as attacker machine)
- VirtualBox
- Text editor (Nano).

3. Lab Setup.

-Host OS: Ubuntu 22.04 LTS (Snort Install)
-Attacker OS: Kali Linux
-Connection: Host-Only Adapter (192.168.56.0/24)
-Network Interface: enp0s3(On ubuntu)

4. Configuration Steps

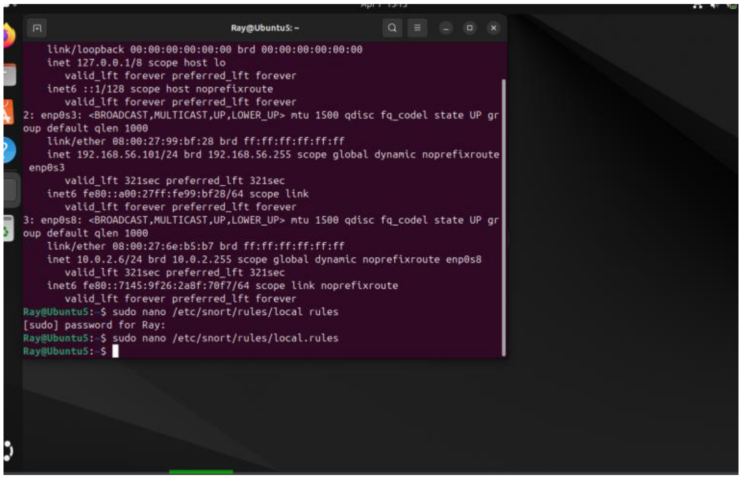
On Ubuntu:

- Installed Snort and Apache2.
- Configured custom Snort rules in /etc/snort/rules/local.rules.
- Ran Snort in console mode for real-time monitoring.

On Kali Linux:
- Used ping to generate ICMP traffic.
- Used curl to simulate HTTP requests.

5. Snort Rules Created

Rule 1: ICMP Detection

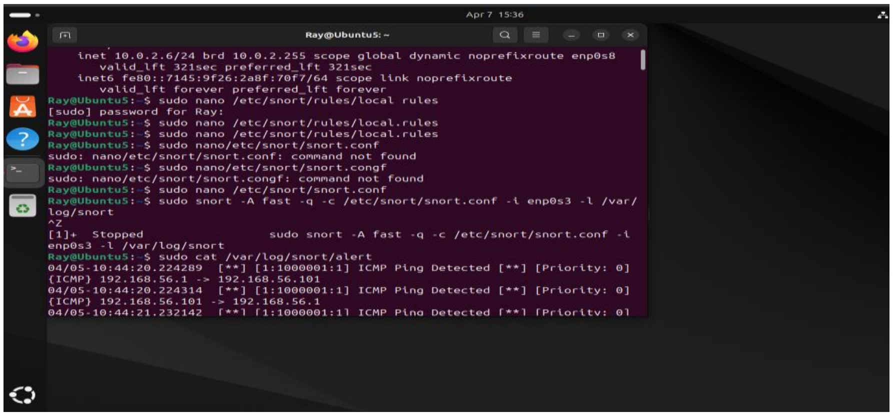


```
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP gr
oup default qlen 1000
    link/ether 08:00:27:99:bf:28 brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.101/24 brd 192.168.56.255 scope global dynamic noprefixroute
 enp0s3
       valid_lft 321sec preferred_lft 321sec
    inet6 fe80::a00:27ff:fe99:bf28/64 scope link
       valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP gr
oup default qlen 1000
    link/ether 08:00:27:6e:b5:b7 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.6/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s8
       valid_lft 321sec preferred_lft 321sec
    inet6 fe80::7145:9f26:2a8f:70f7/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
Ray@Ubuntu5:~$ sudo nano /etc/snort/rules/local rules
[sudo] password for Ray:
Ray@Ubuntu5:~$ sudo nano /etc/snort/rules/local.rules
Ray@Ubuntu5:~$
```

- Add above mention rule by typing cmd sudo nano /etc/snort/rules/local.rules
- And add this [alert ICMP any any -> any any (msg:"ICMP Ping Detected"; sid:1000001; rev:1;)]
- Update snort conf file _ sudo nano /etc/snort/snort.conf
- Find #include $RULE_PATH/local.rules and uncomment it by removing the # in front of it. Save the file, then exit.
- Now, run the command sudo snort -A console -q -c /etc/snort/snort.conf -i enp0s3. Then, run the ping command from the Kali machine to ping the Ubuntu IP address and check if the ICMP packet is detected.

Now we can see snort is printing an alert on the snort console whereas we can check the log file by running sudo snort -A fast -q -c /etc/snort/snort.conf -i enp0s3 -l /var/log/snort now again trigger ping from attackers machine which is kali in our case, then we can see Snort will now write alerts to /var/log/snort/alert.

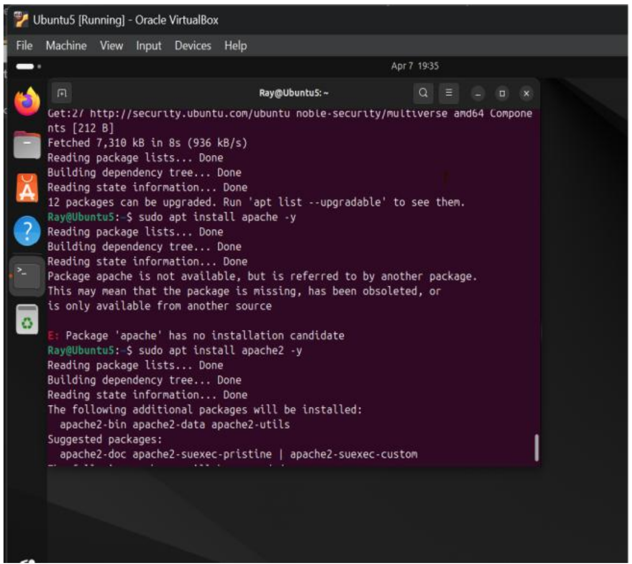
ICMP | ping <Ubuntu-IP> | Alert: ICMP packet detected (Ip=192.168.56.101)



```
    inet 10.0.2.6/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s8
       valid_lft 321sec preferred_lft 321sec
    inet6 fe80::7145:9f26:2a8f:70f7/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
Ray@Ubuntu5:~$ sudo nano /etc/snort/rules/local rules
[sudo] password for Ray:
Ray@Ubuntu5:~$ sudo nano /etc/snort/rules/local.rules
Ray@Ubuntu5:~$ sudo nano /etc/snort/rules/local.rules
Ray@Ubuntu5:~$ sudo nano/etc/snort/snort.conf
sudo: nano/etc/snort/snort.conf: command not found
Ray@Ubuntu5:~$ sudo nano/etc/snort/snort.congf
sudo: nano/etc/snort/snort.congf: command not found
Ray@Ubuntu5:~$ sudo nano /etc/snort/snort.conf
Ray@Ubuntu5:~$ sudo snort -A fast -q -c /etc/snort/snort.conf -i enp0s3 -l /var/
log/snort
^Z
[1]+  Stopped                 sudo snort -A fast -q -c /etc/snort/snort.conf -i
enp0s3 -l /var/log/snort
Ray@Ubuntu5:~$ sudo cat /var/log/snort/alert
04/05-10:44:20.224289  [**] [1:1000001:1] ICMP Ping Detected [**] [Priority: 0]
{ICMP} 192.168.56.1 -> 192.168.56.101
04/05-10:44:20.224314  [**] [1:1000001:1] ICMP Ping Detected [**] [Priority: 0]
{ICMP} 192.168.56.101 -> 192.168.56.1
04/05-10:44:21.232142  [**] [1:1000001:1] ICMP Ping Detected [**] [Priority: 0]
```

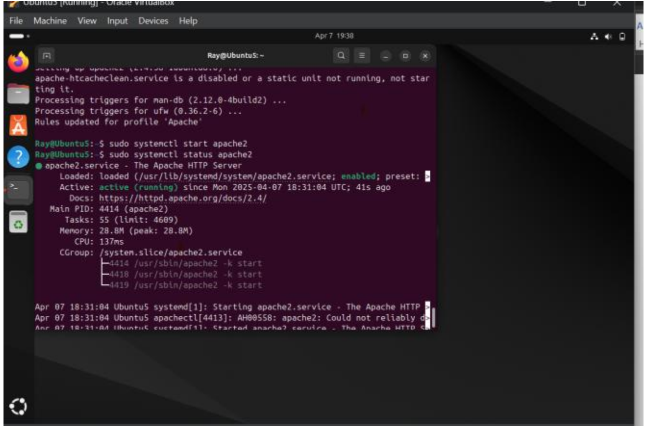
Rule 2: HTTP Detection

Install Apache Server, Follow the same step as above check add the HTTP rule which has 80 port numbers and check the log file which will detect HTTP packets once we generate the traffic from Kali by using [curl http://<Ubuntu-IP>][Curl http://192.168.56.101]. Kindly follow below attached screenshot below:


Ubuntu5 [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Apr 7 19:35
Ray@Ubuntu5: ~

```
Get:27 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Compone
nts [212 B]
Fetched 7,310 kB in 8s (936 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
12 packages can be upgraded. Run 'apt list --upgradable' to see them.
Ray@Ubuntu5:~$ sudo apt install apache -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Package apache is not available, but is referred to by another package.
This may mean that the package is missing, has been obsoleted, or
is only available from another source

E: Package 'apache' has no installation candidate
Ray@Ubuntu5:~$ sudo apt install apache2 -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom
```

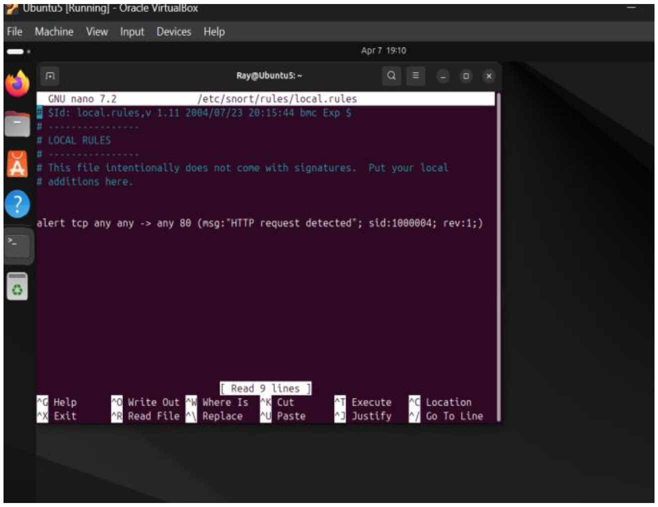

Ubuntu5 [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Apr 7 19:38
Ray@Ubuntu5: ~

```
apache-htcacheclean.service is a disabled or a static unit not running, not star
ting it.
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for ufw (0.36.2-6) ...
Rules updated for profile 'Apache'

Ray@Ubuntu5:~$ sudo systemctl start apache2
Ray@Ubuntu5:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: >
     Active: active (running) since Mon 2025-04-07 18:31:04 UTC; 41s ago
       Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 4414 (apache2)
      Tasks: 55 (limit: 4609)
     Memory: 28.8M (peak: 28.8M)
        CPU: 137ms
     CGroup: /system.slice/apache2.service
             ├─4414 /usr/sbin/apache2 -k start
             ├─4418 /usr/sbin/apache2 -k start
             └─4419 /usr/sbin/apache2 -k start

Apr 07 18:31:04 Ubuntu5 systemd[1]: Starting apache2.service - The Apache HTTP >
Apr 07 18:31:04 Ubuntu5 apachectl[4413]: AH00558: apache2: Could not reliably d>
Apr 07 18:31:04 Ubuntu5 systemd[1]: Started apache2.service - The Apache HTTP S>
```

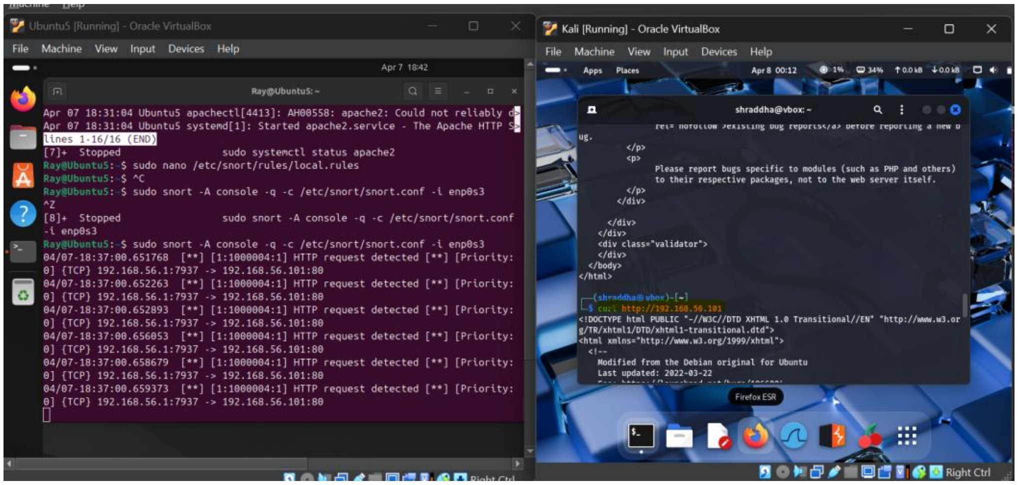
Configured HTTP Rule sudo nano /etc/snort/rules/local.rules



```
  GNU nano 7.2                /etc/snort/rules/local.rules
# $Id: local.rules,v 1.11 2004/07/23 20:15:44 bmc Exp $
# ----------------
# LOCAL RULES
# ----------------
# This file intentionally does not come with signatures.  Put your local
# additions here.


alert tcp any any -> any 80 (msg:"HTTP request detected"; sid:1000004; rev:1;)
```

6. Testing & Results



```
Apr 07 18:31:04 Ubuntu5 apachectl[4413]: AH00558: apache2: Could not reliably d>
Apr 07 18:31:04 Ubuntu5 systemd[1]: Started apache2.service - The Apache HTTP S>
lines 1-16/16 (END)
[7]+  Stopped                 sudo systemctl status apache2
Ray@Ubuntu5:~$ sudo nano /etc/snort/rules/local.rules
Ray@Ubuntu5:~$ ^C
Ray@Ubuntu5:~$ sudo snort -A console -q -c /etc/snort/snort.conf -i enp0s3
^Z
[8]+  Stopped                 sudo snort -A console -q -c /etc/snort/snort.conf
-i enp0s3
Ray@Ubuntu5:~$ sudo snort -A console -q -c /etc/snort/snort.conf -i enp0s3
04/07-18:37:00.651768  [**] [1:1000004:1] HTTP request detected [**] [Priority:
0] {TCP} 192.168.56.1:7937 -> 192.168.56.101:80
04/07-18:37:00.652263  [**] [1:1000004:1] HTTP request detected [**] [Priority:
0] {TCP} 192.168.56.1:7937 -> 192.168.56.101:80
04/07-18:37:00.652893  [**] [1:1000004:1] HTTP request detected [**] [Priority:
0] {TCP} 192.168.56.1:7937 -> 192.168.56.101:80
04/07-18:37:00.656053  [**] [1:1000004:1] HTTP request detected [**] [Priority:
0] {TCP} 192.168.56.1:7937 -> 192.168.56.101:80
04/07-18:37:00.658679  [**] [1:1000004:1] HTTP request detected [**] [Priority:
0] {TCP} 192.168.56.1:7937 -> 192.168.56.101:80
04/07-18:37:00.659373  [**] [1:1000004:1] HTTP request detected [**] [Priority:
0] {TCP} 192.168.56.1:7937 -> 192.168.56.101:80
```

```
        rel="nofollow">existing bug reports</a> before reporting a new b
ug.
          </p>
          <p>
                Please report bugs specific to modules (such as PHP and others)
                to their respective packages, not to the web server itself.
          </p>
        </div>

      </div>
    </div>
    <div class="validator">
    </div>
  </body>
</html>

┌──(shraddha㉿vbox)-[~]
└─$ curl http://192.168.56.101
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.or
g/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <!--
    Modified from the Debian original for Ubuntu
    Last updated: 2022-03-22
```

7. Conclusion

The Snort-based NIDS was successfully configured to detect ICMP and HTTP traffic. Custom rules triggered real-time alerts, demonstrating the effectiveness of Snort in monitoring specific network protocols.